



A Protocol Framework for a Service-Oriented Future Network Architecture

3rd GI/ITG KuVS Fachgespräch Future Internet

B. Reuther, D. Schwerdel, A. Siddiqui, P. Müller

University of Kaiserslautern
ICSY Lab



Goal: short and long term Flexibility

- Goal: Enable adaptivity according to requirements and constraints (short term flexibility)
- Goal: Enable extendability to add, change, and remove protocols (long term flexibility)
- Scope: everything between transport technologies (typical layer-2) and applications



Approach

- Apply principles of SOA
 - A service is an abstract representation of a building block (e.g. micro-protocol)
- Decoupling of building blocks within a node
 - Building blocks should be self-contained
 - Well defined interfaces
- Interaction of building blocks is defined by workflows
 - Workflows should be defined and adapted as needed
- Minimize assumptions about other nodes
 - Deal with heterogeneous nodes
- Decoupling of applications and network functionality
 - Applications request and receive a service, without knowledge about building blocks (“Service-oriented API”)



Steps

- I. Framework for building blocks
 - Manage building blocks and their interaction (execution of workflows)

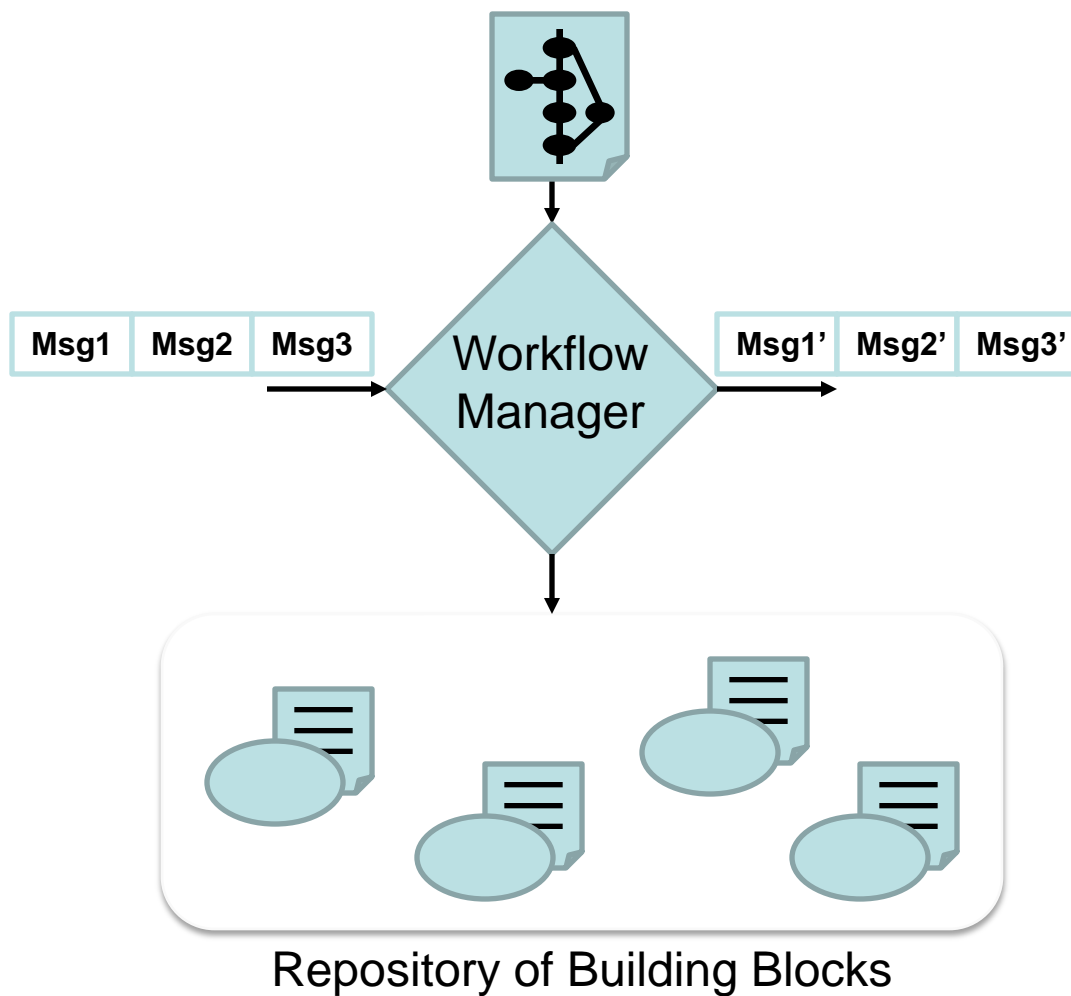
- II. Selection and Composition
 - Define which building blocks are used and how these interact (define workflows)
 - Based on:
 - Application requirements
 - (local) network constraints
 - Descriptions of building blocks

- III. Heterogeneity
 - Adapt to capabilities of other nodes
 - Impact on Selection and Composition

- ...



I. Framework: Workflow execution



Building block and its Description (defined per node)



Workflow description (created / signaled per flow)



Data for a specific BB (exchanged when needed)



I. Framework for building blocks

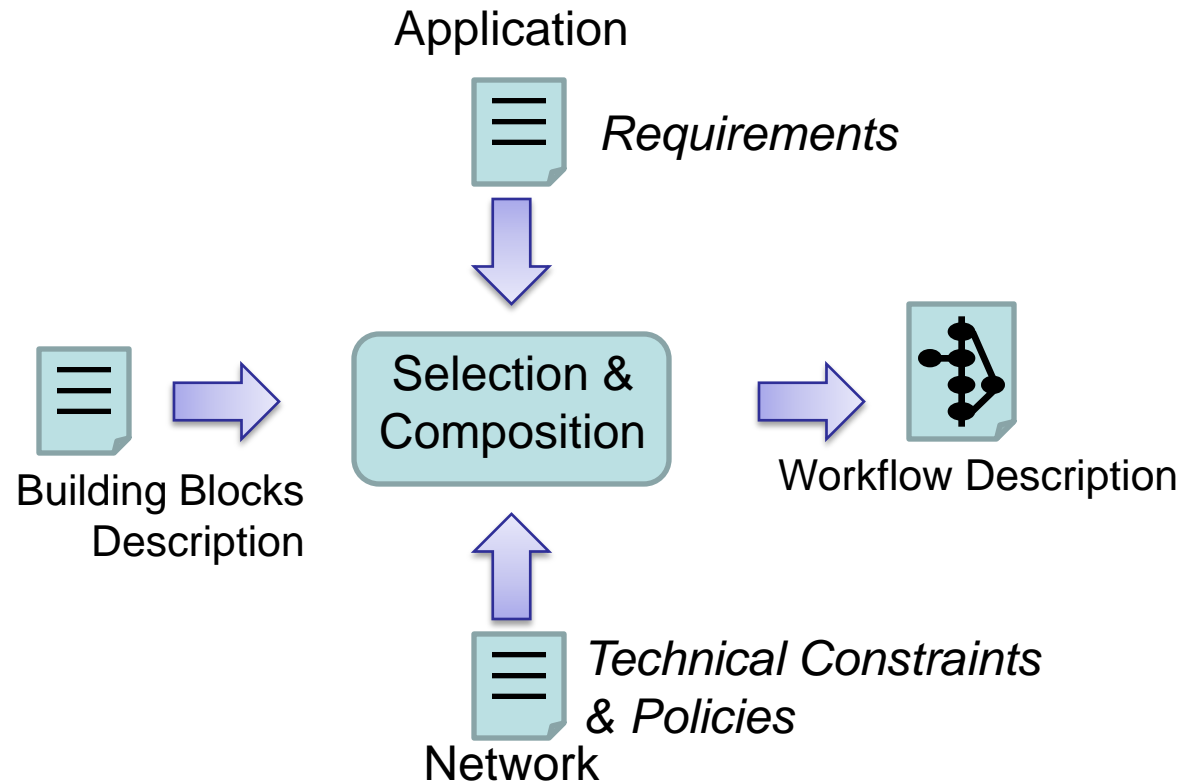
- Framework functionality
 - Workflow execution
 - Infrastructure to support workflow execution
 - Subscribe / publish notifications
 - Timer
 - Shared data structure for exchanging well known values
 - ...
 - Manage repository for building blocks and their description
 - Support connection management



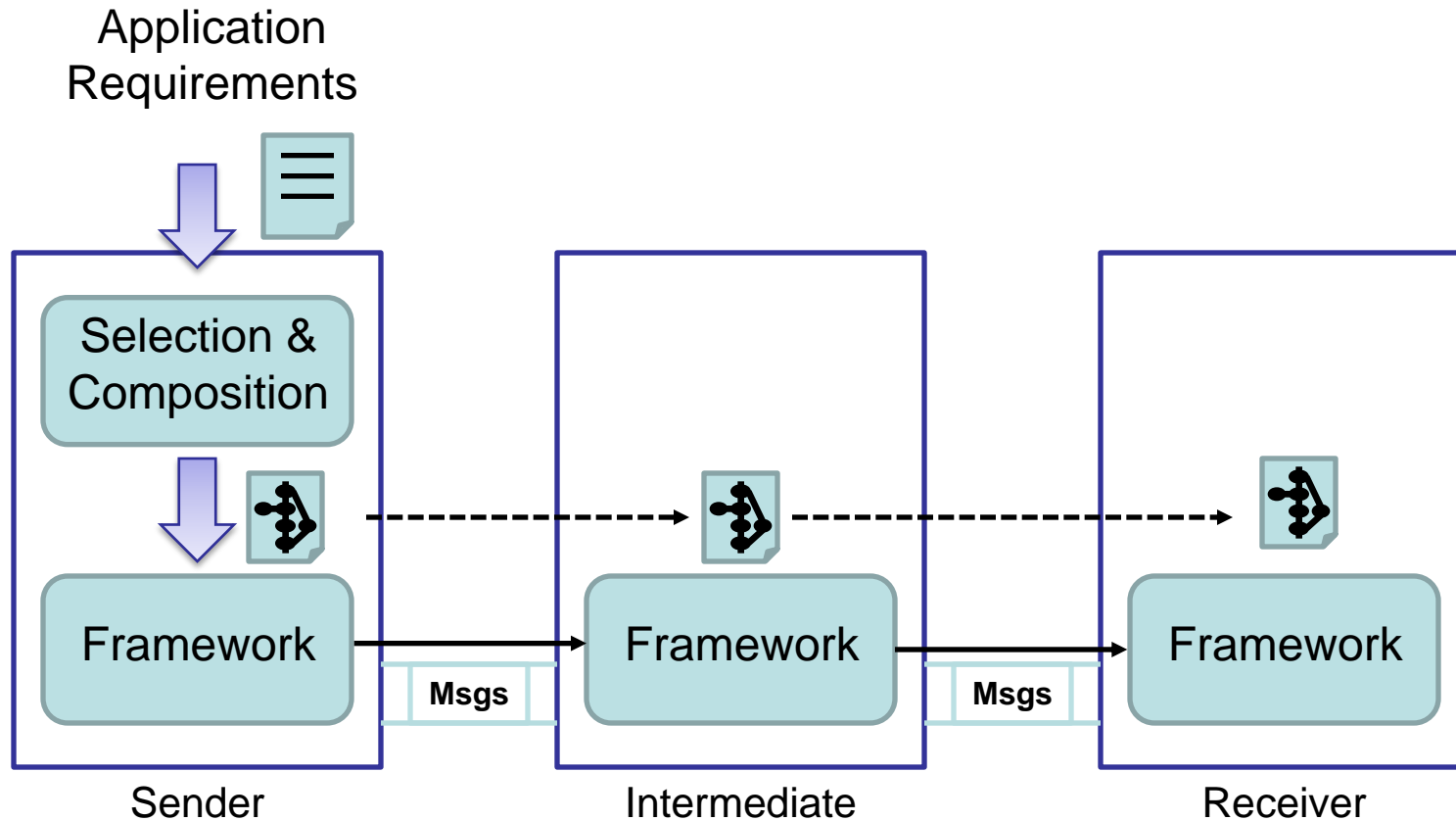
I. Framework for Building Blocks

- Advantages
 - Workflow execution does not depend on hard-coded algorithms or external specifications (e.g. by standard body)
 - Simplifies usage of new building blocks
 - Well-defined interfaces of building blocks support validation and even adaptation of workflows
- Disadvantages
 - Additional overhead
 - Some common basic building blocks are still prerequisite

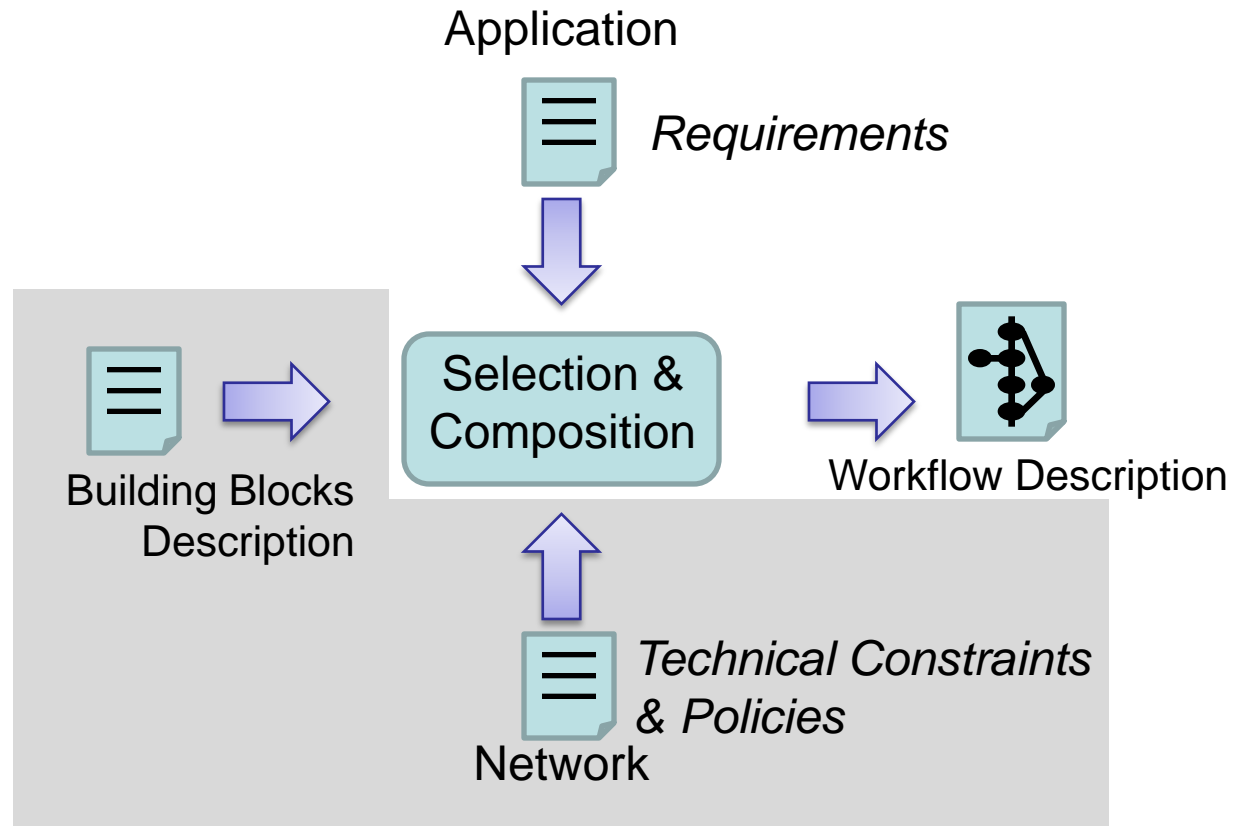
II. Automatic Selection & Composition



II. Signalling the workflow



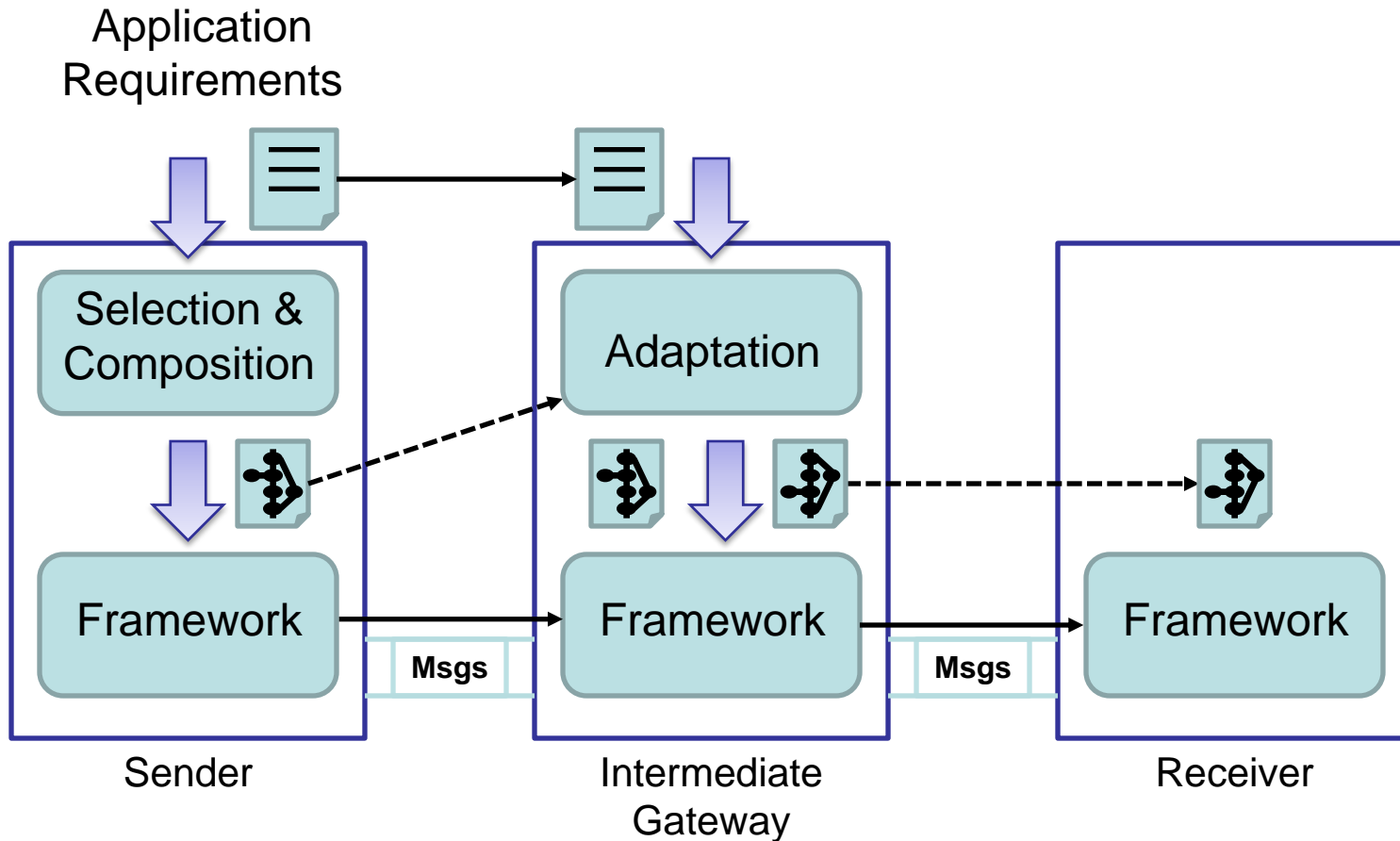
II. Selection & Composition



May vary within the network



II. Adaptation of Workflow



II. Automatic Selection & Composition

- Advantages
 - Dynamic adaptation to changing application requirements and network constraints
 - Usage of new building blocks possible as soon as they are available
 - Handles large number of diversified workflows
- Disadvantages
 - Requires non trivial description of building blocks
 - Might be a time consuming process



III. Heterogeneity

- Heterogeneity: Node A is uncertain about functionality of node B
 - Uncertainty can be tolerated :
 - If a functionality is not mandatory
 - If functionality can be delegated (also back to sender)
 - If few alternatives exist (support all)
 - Negotiation required:
 - „on demand“, lead to additional delay
→ appropriate for delay tolerant flows
 - „a priori“, lead to additional communication overhead
→ appropriate for functionality that changes rarely
 - Impact on initial definition of workflows or adaptation of workflows

